ATLAS NOTE

CERN-PH/2006-000

April 23, 2008

Data Preparation for the ATLAS High-Level Trigger Calorimeter Algorithms

The ATLAS Collaboration

Abstract

This note describes the data preparation layer for the ATLAS High Level Trigger Calorimeter Algorithms. The same infrastructure is relevant to various calorimeter based algorithms (electrons, photons, taus, jets, missing E_T and muons). Fast processing and robustness are fundamental prerequisites for the operation of the trigger reconstruction algorithms.

1 Introduction

Some of the ATLAS sub-detectors data acquisition electronics (the calorimeters and part of the muon systems) were designed to participate in the hardware-based part of the trigger (the level 1 - LVL1) [3–6]. All sub-detectors participate on the software trigger (level 2 - LVL2 and Event Filter - EF) [7]. One important phase for any trigger software algorithm is the data preparation step which provides the conversion of the bytes of data produced by the detector electronics into a convenient form for the trigger algorithms. In the case of the calorimeters, the digital information provided by the detector must be converted into calorimeter cells as input to the reconstruction algorithms. A good data preparation step will provide the input to the trigger software in an organized manner, so that access to the prepared data is optimized. This note describes this step for the calorimeter trigger software. The same software data preparation layer is used for the LVL2 and EF for many different algorithms (electrons, photons, taus, jets and muon algorithms) [8].

1.1 Calorimeters Readout

The LAr calorimeter readout unit is the calorimeter cell. The cell electrodes receive the current due to the drift electrons in the liquid argon and form a triangular shaped signal [3]. The shaping and readout of this signal is performed by the Front-End Electronics. To preserve the dynamic range and the energy resolution, the signal is shaped with 3 possible gains. The Front-End Boards (FEBs) save analog samples of the signals coming from the detector at the bunch crossing rate (25 ns). Each FEB can process up to 128 LAr calorimeter cells.

The signals are converted by the FEBs to digital if the event is accepted by the LVL1 Trigger. The digital information is sent to the ReadOut-Drivers (RODs). These are Digital Signal Processor (DSP) based machines, fast enough to deal with a number of input channels (2 FEBs feed one ROD DSP). From the pulse shape digitized at the FEB, the energy deposited in any cell can be calculated.

Data from one ROD with at most 256 channels are sent to a ReadOut Buffer (ROB). The ROBs keep this data fragment until requested by the LVL2 or by the Event Builder (EB). The EB will request the fragments from the whole detector in case the event is approved by the LVL2 and send it for further processing at the EF farm.

In the Tile Calorimeter case [4], the photons produced by the scintillators are treated by photomultipliers, which produces a negative shaped pulse. The electronic signal already digitized is saved into an on-detector memory waiting for the accept signal from the LVL1 trigger. For each Tile Calorimeter module (in a total of 256 modules), there is a so-called drawer with up to 48 photomultipliers and all the readout electronics inserted in the back of the calorimeter structure.

The analog signals from the detector cells are also summed up by dedicated hardware by detector regions in depth. This way, a coarse granularity version of the calorimeter output can be provided in analog mode to the hardware LVL1 processing. These coarse granularity units are called Trigger Towers (TT). Except for the very forward regions, the TT size is 0.1×0.1 in $\eta \times \phi$. The LVL1 hardware algorithm uses some minimal TT energy and isolation quantities to define a possible egamma candidate. A pointing to the found candidate $\eta \times \phi$ position is sent as a seed for software trigger processing. This seed is used to open a region (usually defined in terms of TT coordinates) called Region of Interest (RoI), where the full detector granularity ca be accessed by the reconstruction algorithms.

2 Data preparation

From a general point of view the data preparation for the LAr and Tile Calorimeters is similar. Figure 1 depicts the global scope of the data preparation for the LVL2 calorimeter algorithm. The extra details of

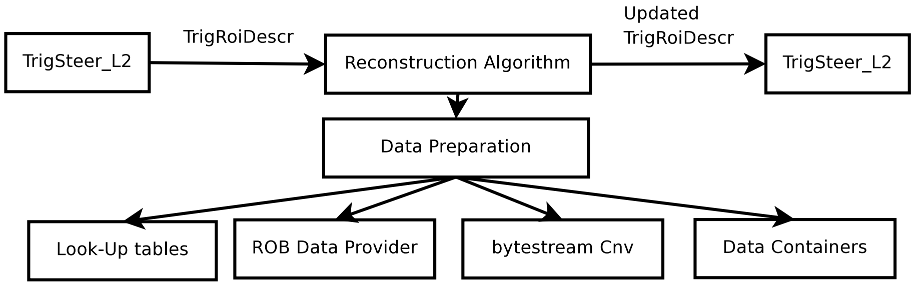

Figure 1: Different parts of the data preparation processing and their relation to the calorimeter algorithm at the LVL2. For details, see text.

the Event Filter data preparation will be detailed later [9].

The LVL2 steering receives the LVL1 information on the acceptance of an event with the RoI η and ϕ coordinates. The reconstruction algorithm gathers a list of ROB identifiers which contain data for a given RoI. Each ROB may partially contain data from TT not pertaining to the RoI (ROB data access is not usually defined by the RoI, but rather by the hardware cabling). An optimal way to map cells to the towers and to the addresses of the ROB must be provided.

The mappings of the ROBs and TT are part of the geometry description and are also used in an offline context. The access to the offline detector description databases which maps any physical position into a set of identifiers are typically very slow, as the full detector description is comprised of a great amount of data. In order to have a faster access, compatible with the LVL2 speed requirements, a look-up table is prepared in the initialization of the algorithm. This table is called the Region Selector.

The ROB identifiers are translated to network addresses of the ROB machines and the data are subsequently requested. The algorithm processing is blocked while the network acquires the data. Thus, running in a multiprocessing environment as the one chosen for the LVL2 may present the advantage to reduce this dead time [7, 10].

When data are received, pointers to the beginning of the different fragments are made available to the data preparation algorithm. Such pointers are passed to the detector specific code (LAr or Tile bytestream conversion codes) which interprets the data format coming from the detector RODs and converts it into an easy to use format (calorimeter cells) for the algorithms.

The last part of the data preparation is to provide the cells in a manner organized for the reconstruction algorithms. For instance, cells are provided by detector layer.

2.1 Data Processing in the Read-Out Drivers

The LAr Digital Signal Processors (DSPs) are able to prepare data in different formats, the most important one being the physics mode. In this mode the DSPs process the nominal 5 samples per cell provided by the front-end electronics. These samples are used to compute the energy deposited in the cell by the particles using an optimal filtering (OF) [11]. This processing is a simple weighted sum of the samples that includes the noise autocorrelation and electronics calibration constants. Also, a normalization factor that converts ADC counts to MeV is included. For cells with energy above a programmable threshold the timing of the signal and the quality of the pulse shape compared to the expectation are calculated. Finally, for each cell, the choice of electronic gain applied to the analog signal in the FEB is also recorded.

Beyond the cell based data, the DSP can also extract global information at an FEB or TT level, which can be used to improve the LVL2 and EF processing speed. The DSP can sums up the energy in a given region in space providing Ex, Ey and Ez totals for these regions. The cell energies are added with these regions using cell position based projection coefficients loaded in the DSP from the database. These quantities can be used to compute jets (at the LVL2) or missing E_T at the LVL2/EF if unpacking the full set of cells data is too time consuming. Providing Ex, Ey, Ez at the TT level is under evaluation to improve jet (LVL2) and missing Et (LVL2 and EF) resolutions.

The pulses from the Tile Calorimeter photomultipliers are also sampled and digitized by 10-bit Analog to Digital Converters (ADCs). During a physics data taking, 7 samples (175ns) of the signal pulses are acquired and transmitted to the RODs. The information is processed using DSPs which also apply optimal filtering energy reconstruction [12]. Again different formats are possible, the main one being the online cell energy reconstruction output.

2.2 Region Selector

As mentioned in the previous sections, to optimize the access to the detector description, part of the information is cached into lookup tables. In the LAr calorimeter case, the information unit to be correlated to the LVL1 position is the Trigger Tower. The $\eta \times \phi$ minimum and maximum of each TT is arranged in a large matrix. Also, the information about the ROD identifier for each TT is included in this table. Multiple tables corresponding to the different calorimeter layers are available. It is possible that a given TT (in the calorimeter crack region) may require data from more than one ROD (one ROD in the Barrel and another in the EndCap). In the Tile Calorimeter case, the geometry information is associated with the calorimeter module identifier and, again, the ROB identifiers. The Look-Up tables with geometry information for LAr and Tile are prepared by accessing the relevant conditions database.

2.3 Data Containers

The data structure of a calorimeter cell includes a part common to LAr and Tile, and parts specifc to both subdetectors. In the software these cells are organized in vectors, each one called a collection. For the Liquid Argon Calorimeter each cell collection holds data for a LAr ROD, corresponding to two FEBs or, at most 256 cells. In the case of the Tile cell collection, there are either 23 cells (in the Barrel) or 13 cells (in the Extended Barrel) per Collection. Data for 4 cell collections are associated to a single ROD. A Tile Calorimeter ROD has data for at most 92 Tile cells. Finally, the Collections are organized in a vector which is called a container.

For the Liquid Argon Calorimeter each cell collection holds data for a LAr ROD, corresponding to two FEBs or, at most 256 cells. In the case of the Tile cell collection, there are either 23 cells (in the Barrel) or 13 cells (in the Extended Barrel) per Collection. Data for 4 cell collections are associated to a single ROD. A Tile Calorimeter ROD has data for at most 92 Tile cells.

The containers for LAr and Tile are stored permanently in memory and the cells and collections are never deleted. This way, on-the-fly memory allocation, which is a typically slow operation in a computing system, is avoided. One problem with reusing collections is that the container must keep track of which collections have already been decoded in a given event. This information is provided by the tools that access the container. If requested another time in the same event, a collection already decoded will not be redecoded.

2.4 ByteStream Conversion

The ROD fragments, containing the energy encoded information are provided to the appropriated bytestream conversion code. Based on the ROD fragment identifier, the corresponding cell collection is requested to the proper container (LAr or Tile containers). Subsequentially, subdetector specific code is used to perform the data unpacking.

The LAr ByteStream conversion code automatically identifies the fragment type using the ROD version encoded in the ByteStream itself. Depending on the detected format, the correct internal infrastructure is selected.

The bytestream conversion unpacks the energy information from the memory block using the format as described in 2.1. It then returns the energy of the cell, the hardware gain and the time and quality information (if available) for each of the ROD fragment channels. The channel number is used as an index to the cell position in the cell collection, so that each LAr channel is associated to a single predefined cell object in the collection. Each cell is updated with the current values of energy, time, quality and hardware gain.

In the unpacking step, typically more cells are requested than those contained in the RoI as data from 1 FEB may extend over several TTs, Furthermore, the trigger reconstruction algorithms require a per layer data access.

This results in a very complex operation with many checks of the cell layer and position. To speed-up the process, some of the geometry information is prepared in the initialisation of a run. During this stage some of the maps between TT identifier and the associated groups of cells are assembled. Using the TT identifier list obtained from the Region Selector, a chain of cells for those TTs can be obtained, simplifying the algorithm code.

The Tile Byte Stream conversion also checks the ROD format via a fragment type identifier in the ByteStream and the correct method to unpack of the data is chosen. The data is decoded and the energy values are stores on a pre-allocated raw data structure. This is again used to avoid online memory allocation. The energy, time and quality are stored together with the ADC identifier for each cell in a Tile Calorimeter drawer. This raw data is copied into the cells. The mapping of raw data to cells is the same for every drawer in a given calorimeter sector. To speed up the processing a mapping is built to the indices of the cells that correspond to each raw data.

Each Tile drawer is unpacked into a cell collection. The data providing in this case is much simpler than in the LAr case. Algorithms are able to iterate through the whole collection after the unpacking is done.

2.5 Data Preparation in the Event Filter

The Data Preparation tools in the EF make use of the same data unpacking approach that is used by the LVL2. However, as the EF has a larger time budget, offline algorithms and tools are used to process the cells. The cells are then stored in one container for all the cells designed for offline reconstruction. This approach was taken to profit from the developments of the offline groups.

For each of the subdetectors (EM, HEC, FCal and Tile), the EF cell container is filled. This provides the possibility of skipping, if needed, one of the calorimeter sections. Once the container has been filled with the corresponding calorimeter cells, a set of tools are executed to check the container, organize it according to the subdetector they belong to, and to perform cell based calibrations.

3 Algorithms and Performance

The HLT selection is divided in two levels (LVL2 and EF) and contains a sequence of feature extraction (FEX - reconstruction algorithms) and hypothesis (HYPO - decision making) algorithms. As an example, the LVL2 electron and photon reconstruction start from the seeded LVL1 RoI and build a cluster object that gathers all the important features for e/gamma identification. The HYPO algorithms use those features to reach a decision, normally by performing simple cuts. To give another example the missing E_T in the EF is calculated using the total vectorial sum of the energy of the calorimeter cells. More details on the feature extraction and hypothesis algorithms can be found in [8].

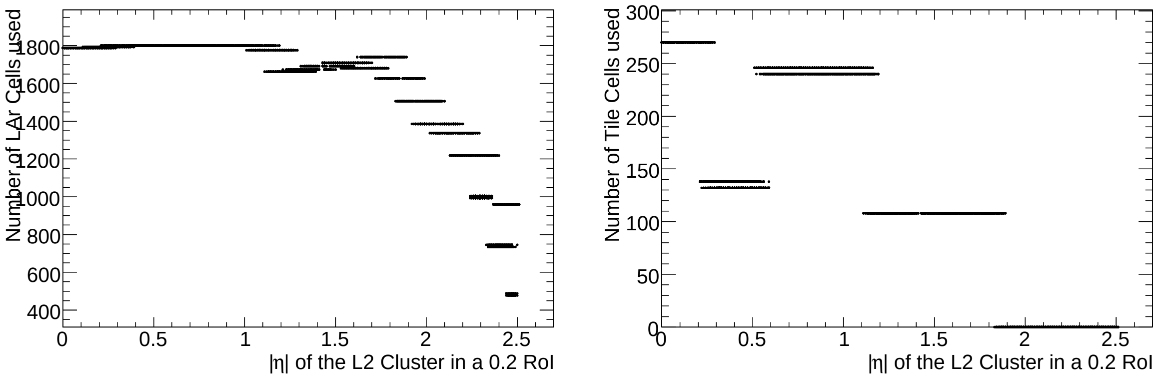

Figure 2: Number of cells used in the T2CaloEgamma Algorithm as a function of η for the LAr (EM, EMEC and HEC) Calorimeters (left) and for the Tile Calorimeter. These distributions were made for the standard LVL2 egamma reconstruction RoI Size (± 0.2).

In the following LVL2 and EF results are treated separately. The results are based on ByteStream (BS) files prepared with a format similar to the ATLAS raw output data. A detail study of the memory footprint and initialization time was performed. A good fraction of the initialization time is taken by the detector geometry assembling, for example, the filling of the cells coordinates and the Region Selector tables. This accesses databases with detector conditions and possibly files with complementary information. Also, for the normal trigger running, it is essential to make sure that the algorithms use a stable amount of memory. This was measure to be the case.

In order to better understand the processing time performance of the algorithms, it is important to know how much data is actually necessary. This can be done checking how many cells are requested on average as a function of η. This is shown in Figure 2. There are separated results for LAr (left) and Tile (right) cells. The distribution on the left shows that the number of used cells in the barrel is quite uniform. Since the granularity is reduced at the EndCaps, less cells have to be unpacked.

The distribution of the number of cells unpacked for the Tile Calorimeter depends on the number determined by the number of drawers to be unpacked. In the very central region ($|\eta| < 0.4$), data from negative and positive rapidities must be accessed to complete the RoI. As a consequence, the number of drawers to be unpacked is doubled. A similar effect happens in the region between the TileCal Barrel and Extended Barrel.

For the standard LVL2 e/γ selection based on a RoI size of $\delta\eta \times \delta\phi = 0.4 \times 0.4$, the algorithm execute time for each of the processing steps can be measured. The results are shown in Figure 3 separately for the EM (left - only LAr access) and for the hadronic (Tile and HEC access). These plots are based on more than 15 thousand events. For the EM part, it can be seen, that even though in the calorimeter crack region (around $\eta = 1.5$), fewer cells are used by the algorithm (see Figure 2), these cells are distributed in more than one ROB (1 ROB from the Barrel and another from the EMEC). This results in a processing time overall increase.

As shown in the Figure, the bytestream conversion makes up a large fraction of the total processing time (about 64%) for the EM part. The rest being used by the algorithm. For the hadronic part, this proportion is much worse (about 90%). The conversion times are especially slow at the regions covered by the Tile Calorimeter and in proportion to the number of Tile Calorimeter modules accessed. Timing optimizations are in progress for this section. The results are also summarized in the Table 1. As in the figure, ROB data fetching times are not included. ROB retrieval times can only be evaluated during real data taking.

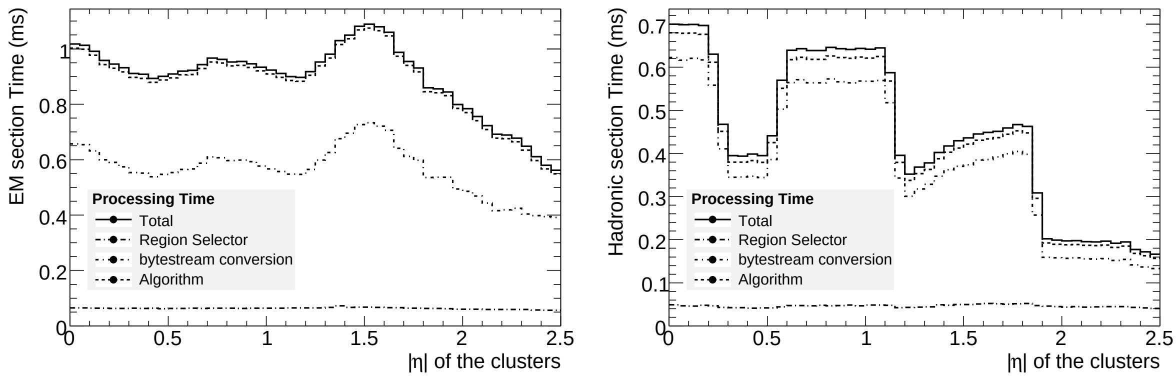

Figure 3: Cummulative time spent in the different phases of the LVL2 e/γ algorithms as a function of η for the Electromagnetic part (left) of the algorithm and for the Hadronic part (right) in a ± 0.2 RoI (a 2.3GHz machine was used to this measurements).

Reco. step	Region Selector	bytestream conversion	Algorithm	Total
EM 2^{nd} layer	29μs	169μs	146μs	347μs
EM 1^{st} layer	13μs	171μs	113μs	301μs
EM total	21μs	158μs	56μs	243μs
Had Total	46μs	334μs	43μs	438μs
Total	109μs (8.2%)	833μs (62.6%)	358μs (26.9%)	1.33 ms

Table 1: Processing time for different algorithm steps and for different actions. Improvements for the Tile Cal Data preparation should be envisaged. Time measurement excludes ROB data retrieval time (a 2.3 GHz machine was used).

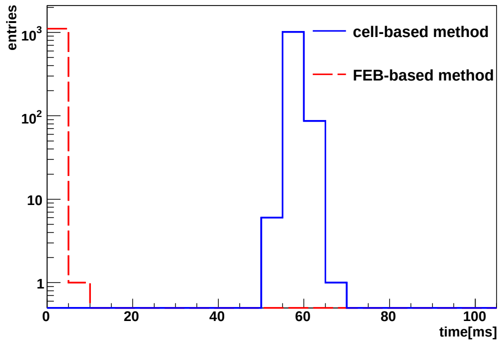

Figure 4: Processing time of the EF MET FEX algorithm. The timing distributions for the cell and FEB method are shown. On average, the time to process the whole calorimeter is 56.6 ms for the cell method and 2.36 ms to the FEB method (an improvement of 24 times). A 2GHz machine was used for this test.

These time measurements indicate that the tile data preparation needs to be improved. Even though, for the barrel region, ~ 6 times fewer cells are accessed, the data preparation time to run the hadronic part is comparable (a proportion about 0.9 to 0.4 at $\eta = 0.5$ - choosing a very good case for Tile) to EM part.

Other algorithms like tau or jets need larger RoI sizes. As an example, the jet algorithm using a RoI size of 1.0×1.0 RoI takes around 10-12 ms.

3.1 Missing E_T in the Event Filter

The Missing E_T EF reconstruction algorithm accesses data from all calorimeters and computes the $E_\mathrm{T}^\mathrm{miss}$ with its E_x, E_y components as well as the total scalar energy sum. In addition, corrections due to energy deposits from muons can be taken into account by including the results from the EF muon reconstruction.

To access the calorimeter data the algorithm uses the same data preparation layer as used by the LVL2. Since the ATLAS calorimeters contain about 200 K calorimeter cells, the access to every single cell can become too time consuming at the trigger level. A faster option is to use the Ex, Ey and Ez energy sums at the FEB level (Section 2.1). So far this method has only been implemented in the LAr data unpacking code, where the impact on the data unpacking time is most significant.

Zero suppression is applied for cells below a given threshold. In the EF Missing E_T reconstruction the two different methods described above have been studied. A faster option uses the FEB information (Energy sums per FEB) from the LAr calorimeter and the cell granularity for the Tile calorimeter. It is worthwhile noting that all the event data is already fetched by the Event Building step, so, no extra network delays are introduced.

In Figure 4 the processing time of the EF Missing E_T FEX algorithm is shown for both methods. If all cells are unpacked, the total processing time peaks around 57 ms, whereas the time has a peak aroun 2 ms if the FEB information is used. This represents a speed-up of a factor of ~ 24.

It was also compared the Missing E_T computation and the total scalar sum for the cells unpacking and the FEB unpacking. The missing E_T calculation shows a similar profile whilst the scalar sum suffers from the effect of the zero suppression. However, due to the drastic improvement in speed if the FEB information, this algorithm is a valid option for the missing E_T reconstruction.

4 Summary and Conclusions

This note describes the implementation of the whole data preparation step for the calorimeter trigger from the detector electronics up to the reconstruction levels. It is fundamental that a data preparation layer is efficient and fast, leaving time for the real physics algorithms. The High-Level Trigger Calorimeter tools described here have been used in the whole physics studying phase of the ATLAS trigger. An unique interface provides access to detector physics quantities (calorimeter cells) obtained with complex computations from the readout data. Knowledge on the detector details is, of course, fundamental to determine the optimal strategy to be followed in this unpacking proceedure. The emphasis of the design approach was to satisfy the important processing time restrictions. They were plainly satisfied as stated in this work. Even for special algorithms, like the missing E_T which process cells from the whole detector, the data preparation performance is still below the required processing interval restrictions. Whenever FEB summary information can be used, significant improvement can be achieved. More optimizations are, anyway, still undergoing.

Also, all the described tools and algorithms have been used during the ATLAS commissioning data taking with cosmic rays. For the moment, this is the only exercise that can emulate the real trigger usage in LHC conditions. Many trigger slices like taus, jets and missing E_T are being successfully explored this way, providing feedback to further improvements of the algorithm developers. This also allow for a detailed study of the data fetching time at the LVL2, which may be a dominant factor for the trigger functioning.

References

[1] ATLAS Collaboration, Atlas Detector and Physics Performancei, Technical Design Report, Internal report, CERN, 1999.

[2] ATLAS Collaboration, The ATLAS experiment at the CERN Large Hadron Collider. Sumbitted to the JINST, Internal report, CERN, 2007.

[3] ATLAS Collaboration, Atlas Liquid Argon Calorimeter Technical Design Report, Internal report, CERN, 1999.

[4] ATLAS Collaboration, Atlas Tile Calorimeter Technical Design Report, Internal report, CERN, 1994.

[5] ATLAS Collaboration, Muon Spectrometer Technical Design Report, Internal report, CERN, 1997.

[6] ATLAS Collaboration, Atlas First Level Trigger Technical Design Report, Internal report, CERN, 1998.

[7] ATLAS Collaboration, Atlas HLT, DAQ and DCS Technical Design Report, Internal report, CERN, 2002.

[8] ATLAS Collaboration, CERN/LHCC **XX** (1900).

[9] D.O. Damazio on bahalf of the ATLAS High-Level Egamma Trigger Calorimeter, ICATPP Conference (2007).

[10] Andre dos Anjos *et al*, IEEE *Trans. Nucl. Sci.* **51** (2004).

[11] W.E. Cleland, E.G. Stern, Nucl. Instrum. Methods **A338** (1994) 467–497.

[12] E. Fullana *et al*, IEEE *Trans. Nucl. Sci.* **53:4** (2006) 2139–2143.